BrightSign Users Guide

Matches:
BrightSign HD2000 Version: 1.3.14
AutoPlay Script Version: 1.3.0.6

California, USA
www.rokulabs.com

INTRODUCTION **4**

BRIGHTSIGN FEATURES **5**

EXAMPLE BRIGHTSIGN SETUPS **7**

EXAMPLES 8
TRY THE DEMO 8
SOFTWARE VERSIONS 9

BRIGHTSIGN DOCUMENTATION **10**

GETTING STARTED **11**

MEDIA TYPES SUPPORTED 12

PLAYING MEDIA WITH BRIGHTSIGN **13**

AUTO PLAYING A VIDEO UPON POWER-UP 14
AUTO PLAYING A PLAYLIST UPON POWER-UP 14
AUTO EXECUTING A ".CSV" FILE UPON POWER-UP 14
AUTO EXECUTING A SCRIPT UPON POWER-UP 15
CREATING A SCHEDULE FOR AUTO PLAYING VIDEOS, STILL IMAGES, PLAYLISTS, OR CSV FILES 15

PLAYLISTS **16**

GETTING STARTED 16
PLAYLIST COMMANDS 17

CREATING INTERACTIVE DISPLAYS **20**

USING EXCEL TO CREATE INTERACTIVE SIGNS 20
STATE FILE (CSV) EXAMPLES 28
ROLLOVER REGIONS USING STATE FILES (CSV) 32
SCHEDULING PLAYBACK (HD2000 ONLY) 35

SCRIPTS **36**

HELLO WORLD 36
ROKU OBJECTS 37
EXAMPLES 37

REGISTRY (HD2000 ONLY) **38**

GPIO CONTROL PORT **38**

LEARNING MORE AND SUPPORT **39**

TROUBLESHOOTING **40**

AUTOPLAY & GENERAL TROUBLESHOOTING 40
PLAYLISTS 41
STATE FILES (CSV) 42
BRIGHTSIGN AND ACCESSORIES 42

APPENDIX **45**

SELECTING COMPACT FLASH CARDS 45
GENERATING HD CONTENT FOR THE BRIGHTSIGN HD2000 USING FINAL CUT PRO 47
DEVICE SOFTWARE (FIRMWARE) AND UPGRADING 48
BRIGHTSIGN SHELL 50
LINUX SHELL (HD2000 ONLY) 54
SETTING THE DATE/TIME ON BRIGHTSIGN (HD2000 ONLY) 54
WRITING BOOT SCRIPT & LOGO TO ROM 54
WIDESCREEN VS. 4X3 DISCUSSION 56

SUPPORTED INPUT DEVICES 56

Introduction

The Roku BrightSigns are complete standalone Digital Sign or Kiosk Controllers that can play content from standard Flash memory cards. BrightSigns have uniquely high quality video playback. This is achieved via professional quality decoders and a scaler that can produce crisp clean 1920x1080 interlaced video from high definition mpeg sources on an HD2000, and 1024x768 video on a VGA monitor from standard definition mpeg sources on an HD600. Still video/pictures can be displayed natively in high definition modes. The BrightSign can be used in looping or interactive modes while displaying video and images on almost any standard or high definition display. It has no moving parts, is completely digital, is compatible with standard media authoring software, and is very reliable. It is intended as a key component of a kiosk, tradeshow display or retail display, and can often be used to replace a PC. Reliability, durability, and flexibility in playing video, audio, and photos make it ideal for other applications as well.

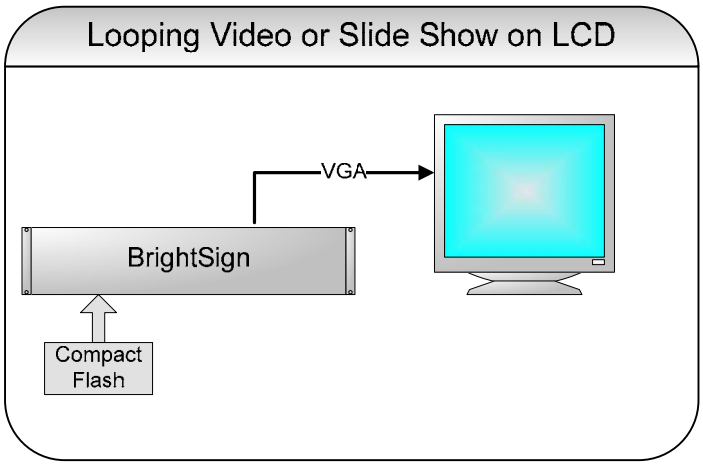
BrightSign Features

	BrightSign HD600	BrightSign HD2000
Display Resolutions		
NTSC	*	
PAL	*	
S-Video (requires adaptor)	*	
640x480 (VGA out)	*	
800x600 (VGA out)	*	
1024x768 (VGA out)	*	*
1280x768 (VGA out)		*
480p (Component out)	*	*
480p (HDMI out)		*
576p (Component out)		*
576p (HDMI out)		*
720p (Component out)	*	*
720p (HDMI out)		*
1080i (Component out)		*
1080i (HDMI out)		*
Video Compression Standards		
MPEG-1	*	*
MPEG-2	*	*
Picture Formats		
BMP	*	*
PNG		*
JPEG		*
Audio Formats		
AC3 pass through	*	*
MP3 decode out to analog and SPDIF	*	*
MP3 decode out to USB	*	*
WAV decode out to analog, USB and S/PDIF		*
5.1 Dolby digital decode (analog and USB output)	*	*
USB Device Support		
Touchscreen	*	*
Speakers	*	*
Mouse	*	*
Keyboard	*	*
Trackball	*	*
Barcode scanner	*	*
USB hub support	*	*
Front Panel		
Power LED	*	*

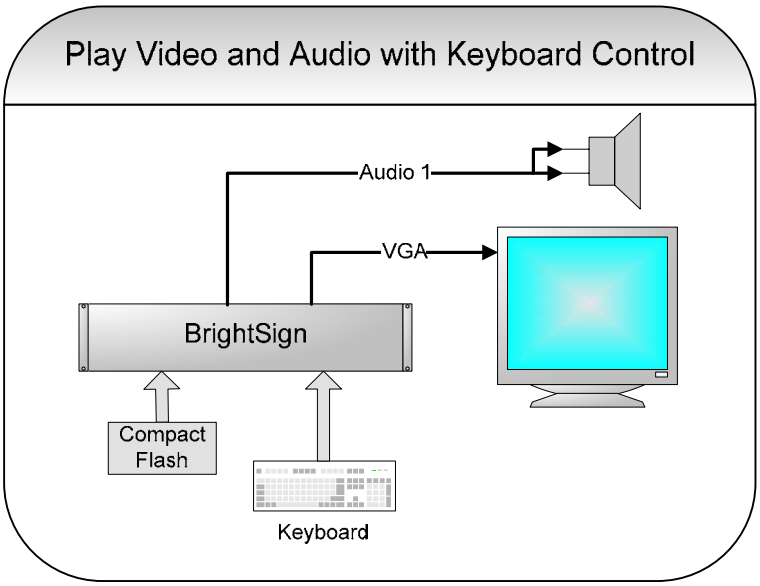
IR receiver	*	*
Compact Flash card slot	*	*
SD/MMC flash card slot		*
Video mode select button	*	*
LEDs: connector activity and video mode	*	*
Back Panel Connectors		
Power	*	*
3 audio out stereo 1/8” mini plugs	*	*
VGA	*	*
Component out 3 RCA jacks: Y-Pr-Pb	*	*
HDMI out		*
2 High Speed USB host ports	*	*
10/100 Ethernet jack	*	*
DB25 control port (for buttons & LEDs)	*	*
RS-232 serial port	*	*
Optical SPDIF audio out	*	*
1/8” 5V serial port		*
8 position dip switch		*
Power		
Typical power consumption	5 watts	10.5 watts
Features		
Auto play and looping video support	*	*
Playlists	*	*
.csv file support	*	*
Scripting	*	*
Real time clock		*
Single files larger than 4GB		*
Upgradable software	*	*
Customer configurable DIP switches		*
Scheduling playback		*
Persistent data storage via eeprom		*
USB mass storage device support		*

Example BrightSign Setups

Probably the simplest way to use BrightSign is to put a video or set of still images on a CompactFlash card, and use a VGA cable to connect the BrightSign to a flat panel monitor. Variations on this setup include using the component out to connect to a High Definition display (like a Plasma TV), or even connecting to a standard definition display. On the HD2000, HDMI out can be used to connect to a display.



As well as looping video or slide shows, you can add audio by connecting powered speakers and you can add interactivity. In the example below, an off-the-shelf USB keyboard is connected to the BrightSign. You can also connect touch screens, buttons and LEDs to the BrightSign for a more custom solution.



Examples

Here are some examples of what you can do with a BrightSign:

- Display a looping video with "DVD+" quality on a variety of monitor types.
- Display a looping slide show in high definition.
- Play looping music.
- Combine any of the above.
- Create interactive displays that play video, photos, or audio in response to customers using touch screens, mouse, keyboard, remote, or a custom button board you have created.
- Create a playlist with Audio, Video, and Images that calls other playlists

Try the Demo

Roku has created a BrightSign demo that shows an example interactive digital sign that uses still slides, video clips, and responds to button, touch screen, or mouse events. It's easy to try if you have a BrightSign. Follow these steps:

1. Upgrade your BrightSign software if needed (see "Software Versions" section)
2. Download the BrightSignDemo.zip file at www.rokulabs.com/brightsign
3. Unzip it and copy the contents into the root directory of a Compact Flash card.
4. Connect your BrightSign to an Elo touch screen, or connect it to a standard monitor and to a USB Mouse or the Roku Button Board.
5. Insert the CF card.
6. Turn on the power.
7. Touch/Click one of the four hot spots, or press button 0 through 3 on the Roku Button Board.

The demo can also be used to try BrightSign on different monitor types and modes. Connect your BrightSign via VGA, the component connection, or HDMI if using the HD2000. Then with the demo running, use the front panel "video mode" button to try the different video modes (480p, 720p, VGA 1024x768, NTSC or PAL). After you put your BrightSign in a specific mode, make sure your monitor/tv supports the mode and verify

your monitor is configured to the correct input type and is connected using the correct cable. You should now see the demo in various video modes, and see how it auto scales and supports widescreen or 4x3.

Note that the front panel "video mode" button is not currently supported on the HD2000.

Software Versions

The version of the BrightSign device software that matches this manual can be found on the cover of the Manual. This version of the software may not have been installed on your BrightSign at the factory (depending on when you purchased it). You should download the latest software from the Roku web site www.rokulabs.com/BrightSign and upgrade your unit by following the steps in this manual in the section "Device Software and Upgrading".

To check your current software version, connect the video and start up the BrightSign without any flash cards inserted. The version number is listed below the Roku logo you'll see on your display. If there's no software version below the logo, your software is older than version 1.1. You can also check which software version is installed on your BrightSign using the shell. See the section in this manual "BrightSign Shell".

In addition to software that resides inside your BrightSign, you will often use Roku's "AutoPlay Script". This script, which you will place on your CF card with your content, is described later in the manual. It also has a version number, and can be downloaded at the same URL as the on device software: www.rokulabs.com/BrightSign.

Roku plans more software updates to add new functions for existing BrightSigns. Watch our web site for more information.

BrightSign Documentation

The complete set of BrightSign documentation is available at www.rokulabs.com/BrightSign. It includes:

- BrightSign Installation and Regulatory Guide
- BrightSign User Guide (this document)
- BrightSign HD600 Hardware Guide
- BrightSign HD2000 Hardware Guide
- BrightSign BrightScript Reference
- BrightSign Object Reference
- BrightSign Specifications
- Example Scripts

Getting Started

Step 1: Get Hardware You Need

To start using the BrightSign, you'll need the following hardware.

- Display – Any monitor, TV, or Elo touch screen with a VGA, Component, Composite (HD600 only), or HDMI (HD2000 only) inputs.
- CF Card Reader/Writer
    - A Compact Flash card to store the demo or your content. Read "Selecting Compact Flash Cards"
    - A USB card reader for your PC or Macintosh that supports compact flash cards. You can get one for less then $20.
- SD Card Reader / Writer
    - Note – SD cards are supported on the HD2000 only. An SD card can be used on the HD2000 in any situation where this document describes CF card usage.
- Optional Accessories
    - USB keyboard
    - USB mouse - You need a USB mouse to use the BrightSign demo if you don't have a GPIO button board or an ELO touch screen.
    - Null Modem cable – You can use this cable to access the BrightSign shell, and to troubleshoot scripts by connecting the BrightSign to your computer's serial port. This is not required to use the BrightSign.
    - SoundBridge Remote

Step 2: Get Latest Software & Autoplay Script

To get the most out of your BrightSign, keep it updated. You can get more information in the "Software versions" section. Please visit www.rokulabs.com/BrightSign, and download the latest updates by clicking on the links in the "Downloads" section:

NOTE: For step by step directions updating your BrightSign software, read the "Device Software and Upgrading" section.

Step 3: Get the Results You Want

Now you're ready to start using the BrightSign. Just add your content and follow the step by step directions in the "Playing Media with BrightSign" section.

Media Types Supported

The following media types are supported:

- **MPEG-2 Video Files (aka "VOB" or "MPG" files)** - These are the same types of files that are found on DVDs. They can be authored using any video editing software such as Pinnacle Studio or iMovie. Video sizes up to 1080i (1920x1080 interlaced) are supported with bit rates up to 25Mbps on the HD2000, and sizes up to D1 (720 x 480) are supported with bit rates up to 9 mbps on the HD600. When a video is played, it will be scaled in real time to the output resolution (which can be much higher definition). The files must contain program streams (transport streams are not supported). Files for the HD600 must have an audio track, and the sampling rate for the audio must be 48K. An audio track is not required in files for the HD2000. For additional information on authoring HD content for the HD2000, see the section in the appendix titled "Generating HD Content for the BrightSign HD2000 using Final Cut Pro."
- **8, 24, 32 bit .BMP files** - These can be authored using almost any paint software such as Adobe PhotoShop, Macromedia Firefly, or even "Paint", which comes with Windows. Any digital camera image can be saved in or converted to this format. A variety of resolutions are supported, up to 1024x768 if outputting via VGA, or 1280x720 (720p) if outputting via "Component".
- **PNG files** (HD2000 only)
- **JPEG files (HD2000 only) –** Resolutions up 1920x1080 are supported. 'Baseline' jpeg files are supported. 'Progressive' jpeg files are not supported.
- **MP3 audio files** – Currently these must be stereo or mono. The HD600 supports files sampled at 48KHZ, while the HD2000 supports files sampled at either 44.1KHZ or 48KHZ. BrightSign has no sample rate conversion. Certain USB speakers only support 48KHz sample rate – when using a USB speaker with this characteristic, all audio will be output at this rate regardless of the sample rate it was recorded at.
- **WAV audio files** – The HD2000 supports WAV files that contain16 bit linear PCM data.
- **Dolby Digital (AC3) 5.1 audio in a video file**
    - BrightSign supports 5.1 AC3 audio playback out of USB and analog outputs (HD600 and HD2000), and SPDIF and HDMI outputs (HD2000 only). For the USB and analog outputs, the AC3 audio is decoded to PCM within BrightSign. This results in 6 tracks over a single USB connector, or three stereo analog connectors. For SPDIF the AC3 audio is sent in its compressed state to the SPDIF receiver. There is also an option within the script control to automatically mix-down the AC3 5.1 audio to stereo PCM.

Playing Media with BrightSign

Media can autoplay from a CompactFlash card upon power-up, play via a script on a CompactFlash card, or be tested using the serial shell commands.

NOTE: On an HD600, video will come out the VGA connector, and the resolution will be 1024x768 unless you have changed this in your .CSV, Playlist, or Script, or by using the front panel video mode button (the video mode button is not currently supported on the HD2000). On an HD2000, video will come out the Component and HDMI connectors and the default video resolution is 720p. If your still image file resolution does not exactly match the display resolution (videomode), then it will be scaled to fit or centered/cropped, depending on the "imagemode" you have selected – see the Playlist Commands section and the Keywords section within the Using Excel to Create Interactive Signs section of the manual for details. Video files will always be scaled to fit.

The following types of media can autoplay:

1. a single MPEG-2 video file (loops)
2. a playlist (also loops). A play list is a simple text file that contains the file names of still slides, video clips, or audio files to play. More information can be found later in this manual under Playlists.
3. a ".CSV" (comma separated values) file that you create using Microsoft Excel (or any csv editor such as OpenOffice or the editor hosted at http://csved.sjfrancke.nl/index.html). BrightSign can use this file format as a description of an interactive sign that responds to buttons, touch screen, mouse, track balls, etc. These are very powerful and easy to create without any scripting. The BrightSign Demo, which is available to download from our web site, uses this method.
4. a custom BrightScript file for arbitrarily complex interactive media sequences.

- **Mpeg Video** - Files must be must be MPEG-2. Files for the HD600 must contain an audio track - if you don't need audio, use "silence".
- **MP3 Audio** - Any audio (in a video or mp3 file) should have a sample rate of 48KHZ (or 44.1KHZ on the HD2000)
- **WAV Audio** – The HD2000 supports WAV files that contain16 bit linear PCM data.
- **Bitmap files** - For still images, use 8-bit, 24-bit, or 32-bit .BMP file format, 1024x768 (or other resolution that matches the video mode you plan to use). Additionally, PNG and JPEG files can be used on the HD2000.

NOTE: 8-bit bitmap files are only supported under HD600 software version 1.1 or later and HD2000 software version 1.0.4 or later. Beginning with Autorun.bas v1.1 and later, Autoplaying a list of images on startup requires the use of a "Playlist" or ".CSV" file.

Auto Playing a Video upon Power-up

To autoplay a video file on power-up or reset, do the following:

1. Copy the video file onto the root directory of a CompactFlash card and name it "autoplay.vob" or "autoplay.mpg".
2. Go to www.rokulabs.com/brightsign and download the "autoplay.zip" file. This contains a script named autorun.bas. Copy autorun.bas into the root directory of your CF card.
3. Insert the CF card into the BrightSign, apply power, and after a few moments, the video should play and auto loop.

Auto Playing a Playlist upon Power-up

A playlist is a text file that describes a sequence of still images, or video clips, or audio clips to play, and the order they should play. See the playlist section of the manual for more detail. To autoplay a playlist file on power-up or reset, do the following:

1. Copy the playlist file onto the root directory of a CompactFlash card and name it "autoplay.bsp". Copy any still graphics, mp3 audio files, or video files referred to by the playlist to the root of the CF card as well.
2. Go to www.rokulabs.com/brightsign and download the "autoplay.zip" file. This contains a script named autorun.bas. Copy autorun.bas into the root directory of your CF card.
3. Insert the CF card into the BrightSign and apply power. After a few moments, the playlist should play and auto loop.

Auto Executing a ".CSV" File Upon Power-Up

BrightSign has a powerful mode whereby it can play a sophisticated sign that is described in a ".CSV" file. CSV stands for "comma separated values", and is a simple text file that contains a table of information. Using Microsoft Excel you can easily create a CSV file that describes a sophisticated interactive sign using a text table. The table describes which media to play, and which media to play when a user presses a button or triggers other events.

To autoplay media described by a .csv file on power-up or reset, do the following:

1. Create your CSV file using Microsoft Excel as described in this manual's section "Using Excel to Create Interactive Signs".
2. Copy your media files and your csv file onto the root directory of a CompactFlash card. Name the file "autoplay.csv".
3. Go to www.rokulabs.com/brightsign and download the "autoplay.zip" file. This contains a script named autorun.bas. Copy autorun.bas into the root directory of your CF card.
4. Insert the CF card into the BrightSign and apply power. After a few moments, the content from the CSV file will play.

Auto Executing a Script Upon Power-Up

Creating a BrightScript file allows you the most control over your interactive digital sign application. To autorun a script:

1. Create the script using a text editor (see the script section of this manual for more information)
2. Save it to the root directory of your CompactFlash card with the name "autorun.bas"

Creating a Schedule for Auto Playing Videos, Still Images, Playlists, or CSV Files

To play videos, still images, playlists, or CSV files at scheduled times, do the following:

1. Create the scheduling file "autoschedule.txt" using a text editor. See the scheduling section of the manual for more detail.
2. Copy "autoschedule.txt" onto the root directory of a CompactFlash card.
3. Copy all video, still image, playlist, and CSV files listed in the "autoschedule.txt" file onto the root directory of a CompactFlash card.
4. Go to www.rokulabs.com/brightsign and download the "autoplay.zip" file. This contains a script named autorun.bas. Copy autorun.bas into the root directory of your CF card.
5. Insert the CF card into the BrightSign, apply power, and after a few moments, the appropriate content will play.

Playlists

A BrightSign playlist is a text file that ends in ".bsp" (BrightSign Playlist). It contains the names of the files to play, in the order they should play. Files are played in sequence, one after the next. This means that it is not currently possible to have a playlist that plays audio and a video at the same time, for example.

Getting Started

To play a playlist, you must have your content, the playlist, and the Roku autoplay script (autorun.bas) on your CF card. Playlists can be played automatically on startup (see the Autoplaying a Playlist section above), or scheduled for playback (see the Creating a Schedule for Auto Playing Videos, Still Images, Playlists, or CSV Files section above).

A simple playlist might contain just these lines:

```
slide1.bmp
slide2.bmp
slide3.bmp
```

This will play the file "slide1.bmp", then "slide2.bmp", then "slide3.bmp", then it will loop.

The files don't have to be bitmaps: they can also be video or audio files. You can also mix media types in a single playlist, and trigger GPIO pins in a playlist.

As well as filenames, there is a set of about 16 commands that can be in a BrightSign playlist. For example, the slideshow playlist above could be modified like this:

```
slideinterval 5
slide1.bmp
slide2.bmp
slide3.bmp
```

Now the playlist will pause 5 seconds between bmp displays.

```
slideinterval 5
slide1.bmp
slide2.bmp
slideinterval 2
slide3.bmp
slide4.bmp
```

Now, the playlist will pause five seconds between slides 1 and 2, and two seconds between slides 3 and 4.

An example/demo playlist can be downloaded from our website:
www.rokulabs.com/BrightSign

Playlist Commands

Here is the list of commands that can appear in a playlist:

.bmp, .mp3, .mpg, .vob, .png (HD2000 only), .jpg (HD2000 only)

Any line that is a file name that ends in one of these extensions is played. Names may contain spaces.

.bsp

You can call another playlist from within a playlist putting the playlist filename on a line by itself. For example: mysubplaylist.bsp. This “sub playlist” will complete one loop and then return control to the main playlist.

debug

Turns on debug information that appears on the console (serial port)

rem <comment>

Any line starting with rem is ignored. Rem is short for “remark”

print <line to print to console>

Prints a line to the console. This is useful for debugging.

audiomode # *(default is 1)*

Sets the Audio Mode. BrightSign has 5.1 audio decode and is controlled via the audiomode. If your audio source is a video file, you can use the "audiomode" commands:

- audiomode 0 5.1 surround sound decode (via USB or all six analog outs)
- audiomode 1 5.1 is mixed down to stereo
- audiomode 2 mute audio

audiooutput # *(default is 0)*

BrightSign has 5 different audio outputs. If none are selected the default is analog audio 1 in stereo. Of the 5 outputs, there are five “audiooutput” modes. The mode is set with the "audiooutput" command. It can have one of these values:

- 0 - Analog audio
- 1 - USB audio
- 2 - SPDIF audio, stereo PCM
- 3 - SPDIF audio, raw AC3
- 4 - analog audio with SPDIF mirroring raw AC3

audiochan # *(default is 2)*

Since there are three analog outs, if analog is selected you must also use the "audiochan" cmd:

- audiochan 2 -- output 1 (this is not a typo)
- audiochan 1 -- output 2
- audiochan 0 -- output 3

volume # *(default is 100)*

Sets the volume to a value between 0 and 100. For example: volume 100

videomode <mode> *(default is 1024x768x60p)*

Set the output framebuffer format. The options are:

VGA:

- 1024x768x60p
- 640x480x60p – HD600 only
- 800x600x60p – HD600 only
- 1280x768x60p – HD2000 only

Component:

- 720x480x60p
- 720x576x50p – HD2000 only
- 1280x720x60p
- 1280x720x50p – HD2000 only
- 1920x1080x60i – HD2000 only

Composite:

- NTSC – HD600 only
- PAL – HD600 only

imagemode # *(default is 0)*

- imagemode 0 – Display the image centered without scaling (may result in cropping if the image is larger than the current videomode resolution).
- imagemode 1 - Scaled to fit. The whole image is displayed as large as possible with correct aspect ratio.
- imagemode 2 - Scaled to fill. Image is scaled to fill screen whilst maintaining aspect ratio
- imagemode 3 - Scaled to fill without maintaining aspect ratio

slideinterval # *(default is 3)*

Sets the number of seconds between slides. Must be an integer. For example: slideinterval 3

lighton

lightoff

These two commands are used to set or clear one of the GPIO outputs. The number can be any integer between 17 and 22. For example: lighton 17 sets LED 0 on the button board.

waitbuttonany

This will pause until any of the GPIO buttons is pressed.

pause

This will pause for the specified number of seconds. The number must be an integer. For example: pause 5

displayregistry *or*

displayregistry-<registry section name>

This will cause the contents of the registry to be displayed on the attached monitor. The optional ‘registry section name’ parameter displays the specified subset of the registry. See the Registry section for more information.

stop

Stops the playlist playback. An example where this might be used is when displaying the contents of the registry.

Creating Interactive Displays

With BrightSign you can create displays that respond to user input. For example, give the customer options of watching different videos depending on which button is pressed. To accomplish this with BrightSign, you need to decide what the input device will be, and you will need to either (a) use Microsoft Excel to create a control table and use BrightSign's "autoplay .CSV files" mode, or (b) create a custom control script.

Using Excel to Create Interactive Signs

This mode is implemented using the "autoplay" script available on Roku's web site. See the prior section "Auto Executing a ".CSV" File Upon Power-Up" for more information.

BrightSign has a powerful mode whereby it can play a sophisticated sign that is described in a ".CSV" file. CSV stands for "comma separated values", and is a simple text file that contains a table of information. Using Microsoft Excel you can create a CSV file that describes a sophisticated interactive sign using a text table. The table describes which media to play, and which media to play when a user presses a button or triggers other events.

A good example of a CSV file is the one used in the Roku BrightSign demo. Run the demo as described in the introduction. Then find the file "autoplay.csv" in the zip file. Copy it out of the zip file onto your desktop, then double click it to load it into Microsoft Excel (you must have Excel installed on your computer. If not, you can load it into any text editor, but it won't be displayed in the nice tabular format). After you edit the file, you can save it with "Save As…" and select "CSV" as the file format to save as.

Below is a section of the autoplay.csv file from the Roku demo. Due to space limitations on the page, it does not include all the columns. The extra columns (not seen below) to the right are similar to the "elo" columns, and describe what happens when buttons are pressed, or when a video finishes or a slide times out.

VIDEOMODE	1024x768x60p					
IMAGEMODE	1					
EVENTS		elo:110:180:350:200	elo:510:180:350:200	elo:110:400:350:200	elo:510:400:350:200	elo:0:0:1024:768
STATE	demo_menu.bmp	Dalmations.mpg	art1.bmp	cars.mpg	bs1.bmp	
STATE	Dalmations.mpg					demo_menu.bmp
STATE	Cars.mpg					demo_menu.bmp
STATE	art1.bmp					demo_menu.bmp
STATE	art2.bmp					demo_menu.bmp
STATE	art3.bmp					demo_menu.bmp
STATE	art4.bmp					demo_menu.bmp
STATE	bs1.bmp				bs2.bmp	
STATE	bs2.bmp				bs3.bmp	
STATE	bs3.bmp				bs4.bmp	
STATE	bs4.bmp				bs5.bmp	

STATE	bs5.bmp				bs6.bmp	
STATE	bs6.bmp				bs7.bmp	
STATE	bs7.bmp				bs8.bmp	
STATE	bs8.bmp				bs9.bmp	
STATE	bs9.bmp				bs10.bmp	
STATE	bs10.bmp				demo_menu .bmp	

Keywords

The first two lines use the optional "videomode" and "imagemode" keywords. These items are optional, but if used, can have these parameters (in the column after the keyword VIDEOMODE or IMAGEMODE).

VIDEOMODE – HD600

1) ntsc
2) pal
3) 640x480x60p
4) 800x600x60p
5) 1024x768x60p
6) 720x480x60p
7) 1280x720x60p

Note: If you set the VIDEOMODE, the front panel button will be locked out. If you leave this option off, the BrightSign will default to 1024x768 VGA, and you can use the front panel button to dynamically change the video mode.

VIDEOMODE – HD2000

1) 720x480x60p
2) 720x576x50p
3) 1024x768x60p
4) 1280x768x60p
5) 1280x720x60p
6) 1280x720x50p
7) 1920x1080x60i

Note: The front panel button is not currently read on the HD2000. If a video mode different from the current video mode is set in a script, the unit will reboot, setting the new video mode to what was specified in the script.

IMAGEMODE

- 0 - Center image. No scaling takes place, only cropping if the image is bigger than the screen.
- 1 - Scale to fit. The image is scaled so that it is fully viewable with its aspect ratio maintained. NOTE: In the 1.0.72 software there is an issue that causes this to fail on "large" images.

- 2 - Scale to fill and crop. The image is scaled so that it totally fills the screen, though with its aspect ratio maintained.
- 3 - Scale to fill. The image is stretched so that it fills the screen and the whole image is viewable. This means that the aspect ratio will not be maintained if it is different to that of the current screen resolution.

VIDEOPLAYERAUDIOMODE <audio mode> (HD2000 only)
Sets the audio mode used when playing a video
- 0 – AC3 Surround
- 1 – AC3 mixed down to stereo
- 2 – No audio

VIDEOPLAYERAUDIOOUTPUT <audio output> (HD2000 only)
Sets the audio output when playing a video
- 0 – Analog audio
- 1 – USB audio
- 2 – SPDIF audio, stereo PCM
- 3 – SPDIF audio, raw AC3
- 4 – analog audio with SPDIF mirroring raw AC3

VIDEOPLAYERAUDIOSTEREOMAPPING <stereo mapping> (HD2000 only)
Selects the analog output when playing video if audio output is 0
- 0 – Stereo audio is mapped to Audio 3
- 1 – Stereo audio is mapped to Audio 2
- 2 – Stereo audio is mapped to Audio 1

VIDEOPLAYERAUDIOVOLUME <volume> (HD2000 only)
Sets the volume when playing video. The volume parameter should be between 0 and 100.

AUDIOPLAYERAUDIOOUTPUT <audio output> (HD2000 only)
Sets the audio output when playing an audio file
- 0 – Analog audio
- 1 – USB audio
- 2 – SPDIF audio, stereo PCM
- 3 – SPDIF audio, raw AC3
- 4 – analog audio with SPDIF mirroring raw AC3

AUDIOPLAYERAUDIOSTEREOMAPPING <stereo mapping> (HD2000 only)
Selects the analog output when playing audio if audio output is 0
- 0 – Stereo audio is mapped to Audio 3
- 1 – Stereo audio is mapped to Audio 2
- 2 – Stereo audio is mapped to Audio 1

AUDIOPLAYERAUDIOVOLUME <volume> (HD2000 only)

Sets the volume when playing an audio file. The volume parameter should be between 0 and 100.

FLIPELO
This keyword isn't used in the example above. The FLIPELO keyword should only be used with a touch screen if the touch screen coordinates appear to be reversed.

```
FLIPELO
IMAGEMODE          2

EVENTS                    elo:110:180:355:210
```

***REGISTRYSECTION* <registry section name> (HD2000 only)**
Directs that any writes to the registry in this CSV file will be written to the specified registry section. If the registry section name is not specified in the CSV file, registry writes will occur in the default registry section named "User". See the Registry section later in the document.

***DELETEREGISTRYSECTION <registry section name>* (HD2000 only)**
Deletes the specified registry section and all registry keys and values within this section.

***DELETEREGISTRYKEY* <registry key> (HD2000 only)**
Deletes the specified registry key from the current registry section. The corresponding registry value is also deleted.

EVENTS
The first line of the CSV file after the keywords is the EVENTS row. For every event type you want to respond to, you must make an entry in this row. You do not need to enter events you don't use. The following event types are supported:

- **elo** - USB Mouse or Elo Touch Screen presses. You define a rectangular touch area using this format:
  - elo:x:y:width:height – based on a 1024 x 768 resolution
  - x:y - represents the starting position, where 0,0 is in the upper left
  - width:height – represent the width and height of the area in pixels

  You can have as many elo events as you wish. Events can overlap touch areas, but in this case, you should only use one at a time (see the demo example and how it uses a touch area that covers the entire screen when not in the main menu).

  Note: You can also use **elor** instead of elo. They are interchangeable.

- **eloc** - USB Mouse or Elo Touch Screen presses. You define a circular touch area using this format:
  - eloc:x:y:radius
  - x:y – represents the center of the circular region
  - radius – radius of the circular region, in pixels

- **button** - Triggers on a GPIO Control Port button press. The format is: button:number, where number is between 0 and 12.
- **timeout** - When a video or slide is displaying, or an audio is playing, you can use a "timeout" event. The format is: `timeout:seconds` where seconds is the number of seconds after the video/image/audio starts that the event will trigger. This event can be used to advance a slide show to the next slide, or exit a looping video after a certain number of seconds of playback.
- **mstimeout (HD2000 only)** – Like 'timeout' above, but the format is timeout:milliseconds where milliseconds is the number of milliseconds after the video/audio/image starts that the event will trigger. Additionally, mstimeout differs from timeout in that while there can only be a single timeout event per CSV file, multiple mstimeout events are supported.
- **videoend** - This event triggers when a video finishes playing. It can be used to move on to another state after the video ends. The format is: `videoend`. There are no parameters.
- **audioend** – This event triggers when an mp3 file finishes playing. It can be used to move on to another state after the audio ends. The format is: `audioend.` There are no parameters.

NOTE: See the "Rollover Regions" section for two additional events you can use with CSV files.

STATE

After the EVENTS row, there are one or more STATE rows. These begin with the keyword STATE. The first cell after the STATE keyword is the name of the video or image file to display, or audio file to play. When the "state is entered", this file starts playing. The remaining cells describe what state to jump to when the corresponding event happens. For example, in the above example, the first state is "demo_menu.bmp". This is the main menu for the demo, and is displayed when the demo starts. While "in this state" (the demo_menu.bmp is showing), if the user presses the elo touch screen or clicks the mouse in the right spot to trigger the "elo:110:180:350:400" event, then the new state "cars.mpg" is jumped to. If you look at the STATE rows, you will find one that starts with "cars.mpg" (there can be only one state with a particular file).

To understand this better, look at the actual demo "autoplay.csv" file in Excel while playing the demo on your BrightSign. As you select options on the BrightSign, see how the .CSV file is telling the BrightSign what to do. You can then edit the "autoplay.csv" file in Excel and Save As… a CSV file to create your own applications.

Writing to the Registry, Displaying Registry Contents, and Copying Registry Contents from a CSV file (HD2000 only)

When using CSV files on the HD2000, it is possible to track the number of times that the system transitions from one state to another using the registry (see the Registry section

later in the document). To do this, append the following to the state name specified in an event column:

:counter(<registry key name>)

In the example above, the state cars.mpg is in the “elo:110:180:350:400” event column. To track the number of times that the user activates this transition, enter “cars.mpg:counter(TouchToDemoMenu) in place of “cars.mpg”. The value in the registry corresponding to the TouchToDemoMenu key is incremented when the corresponding event is triggered.

To display the contents of the registry from a CSV file, use the reserved state name “DisplayRegistry”. That is, add a row to the table with the word “DisplayRegistry” in the column to the right of the keyword STATE. The remainder of the row is like any other row in the table – it contains other states to transition to when events occur. Elsewhere in the table, add the state DisplayRegistry in the column for the event intended to cause the registry to be displayed.

When the system transitions to this state, playback from the prior state is stopped and the contents of the registry are displayed on the monitor attached to the BrightSign.

To display only a subset of the registry, use the following syntax:
DisplayRegistry-<registry section name>

To copy the contents of the registry to a file on a CF card, SD card, or USB mass storage device, enter the following in the state table:
Execute(CopyRegistry <file-name>)
Examples of <file-name> include:
registry-output.csv
sd:/registry-output.csv
usb1:/registry-output.csv

The contents of the registry are written in a comma separated value format, allowing easy import into spreadsheet applications.

The entry “Execute(CopyRegistry <file-name>)” should be placed in the CSV file in the appropriate row and column (appropriate state / event). Unlike other entries in this part of the state table, this entry only executes the specified command (CopyRegistry) and does not transition to a new state.

Controlling GPIO Outputs from a CSV File (HD2000 only)

When using CSV files on the HD2000, it is possible to control the GPIO outputs from a CSV file. GPIO’s can be set by appending a GPIO command to the state specifications in the second column of CSV files. The GPIO’s are set based on the appended command when the state is entered.

The general syntax is:
<state name>:<gpio command>(<gpio_num>;<gpio_num>;gpio_num)
That is, the state name followed by a colon, then the gpio command, then a list of semicolon separated gpio's inside of parentheses.

Gpio 0 corresponds to the first LED on the button board

The GPIO commands are

- gpiosOn – turns on the specified GPIO's; unspecified GPIO's are unchanged.
- gpiosOff – turns off the specified GPIO's; unspecified GPIO's are unchanged.
- gpios – turns on the specified GPIO's; unspecified GPIO's are turned off.

Examples

- Demo_menu.bmp:gpios(1;3;5)
  - LED's 1, 3, and 5 are turned on; the other LED's are turned off when the demo_menu.bmp state is entered
- aq.mpg:gpiosOn(2)
  - LED 2 is turned on; other LED's are unaffected when the aq.mpg state is entered
- ppt10.bmp:gpiosOff(0;2;4)
  - LED's 0, 2, and 4 are turned off; other LED's are unaffected when the ppt10.bmp state is entered
- product1.bmp:gpios()
  - All LED's are turned off when the product1.bmp state is entered

Preloading Images

The autoplay script tries to be smart about preloading still images so that when a user triggers their display, they pop up immediately. Here are the rules that are used to preload an image:

1. If a state (row in the table) only has one possible next image state, then it is preloaded.
2. If a state has more than one possible next image state, but one of them is the next row in the state table, then it (next state row's file) is preloaded. So if you have a sequence of images that display one after the other, put them in sequential rows in the state table to trigger preloading. As an example of this, see the bs1.bmp, bs2.bmp, etc states (rows) in the prior table.

Multiple State Machines within a single CSV file (HD2000 only)

On an HD2000, it's possible to create interactive displays with multiple state machines within a single CSV file. For example, it is possible to create a display which uses a set of inputs and uses HDMI for audio/video output while a different set of inputs controls other content using BrightSign's analog audio outputs. The two state machines can run simultaneously and independently.

To use a second state machine, use the keyword NEXT in the first column of the CSV file following all of the STATE's from the first state machine. Keywords can then be added, followed by the states for the second state machine.

Subsequent state machines can be added in the same way, up to a maximum of four state machines total.

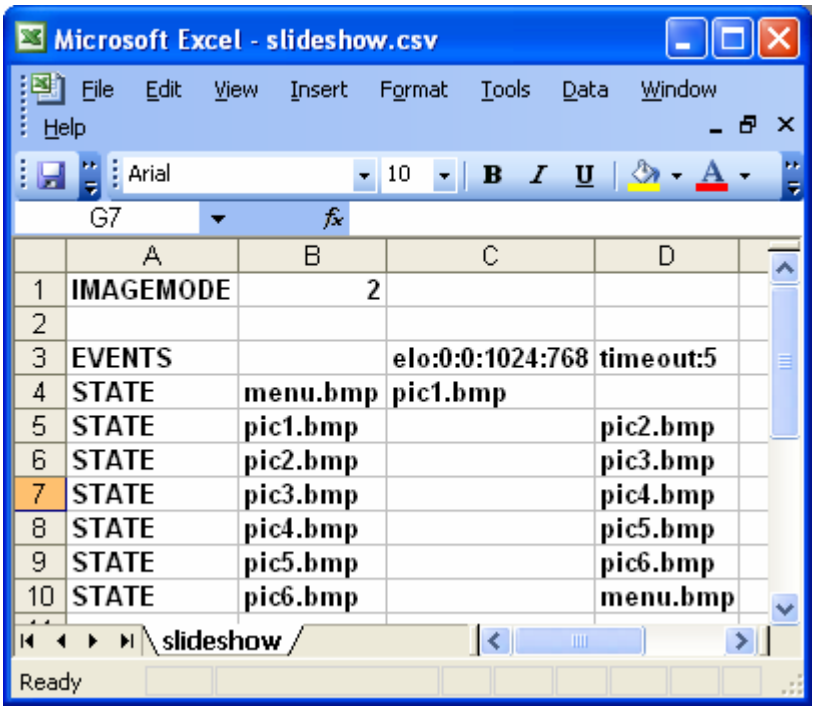
When using multiple state machines, the use of video and images must be specified in the first state machine only. Additionally, the state machines must use independent resources. For example, if state machine 1 is using analog output Audio 1, then it is unavailable to other state machines – therefore, another state machine would use analog output Audio 2.

See Example 4 in the next section for an example of a CSV file with multiple state machines.

State File (CSV) Examples

The BrightSign Demo described above shows that you can design powerful interactive signs using easy to understand CSV files. The following examples further illustrate how easy CSV files are to use.

Example 1: *Slideshow that advances automatically after the first slide plays*



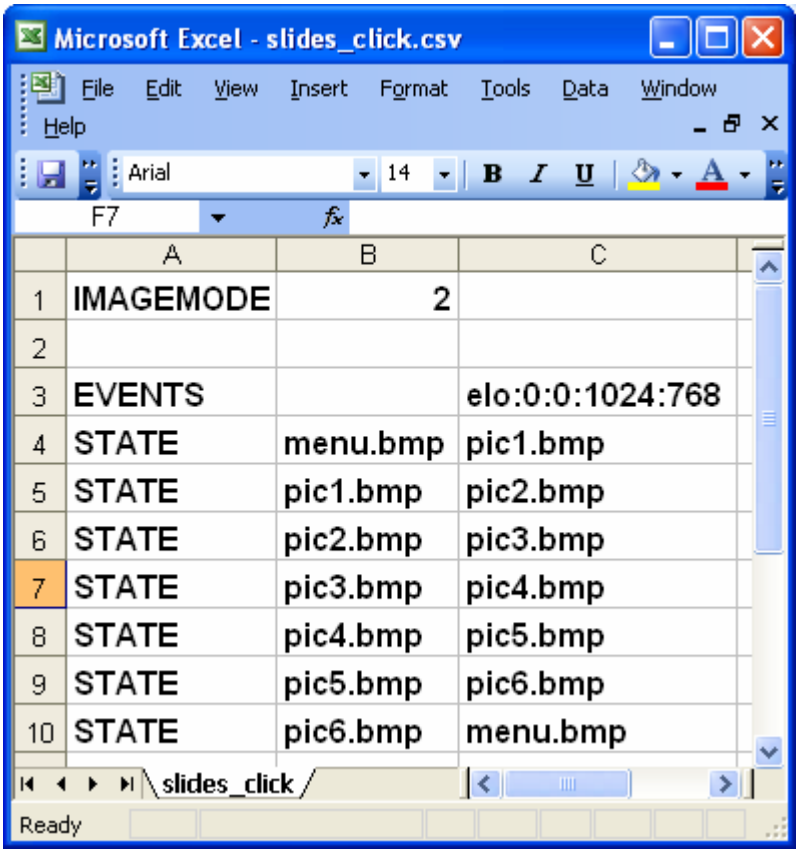
	A	B	C	D
1	IMAGEMODE	2		
2				
3	EVENTS		elo:0:0:1024:768	timeout:5
4	STATE	menu.bmp	pic1.bmp	
5	STATE	pic1.bmp		pic2.bmp
6	STATE	pic2.bmp		pic3.bmp
7	STATE	pic3.bmp		pic4.bmp
8	STATE	pic4.bmp		pic5.bmp
9	STATE	pic5.bmp		pic6.bmp
10	STATE	pic6.bmp		menu.bmp

Example 1 contains two events, an **elo** event (used for ELO touch screens or any Mouse input) and a **timeout** event. The clickable/touch area is defined as the whole screen. One touch or click starts the slideshow.

When this demo starts, **menu.bmp** is displayed since it's in the first STATE position.

- There's no **timeout** entry for **menu.bmp**. If there's no mouse click or touch, **menu.bmp** will remain onscreen forever.
- There's one touch / click event listed for **menu.bmp**. If **menu.bmp** is currently being displayed, and there's a mouse click or touch anywhere onscreen, **pic1.bmp** will be displayed.
- Pictures 1 – 6 all have **timeout** entries. Each image is displayed for 5 seconds before the next image plays. The **timeout** entry for the last image is **menu.bmp**.
- There's no **elo** entry for pictures 1 – 6. So, if there's a mouse or touch event while the images are being displayed, nothing happens until the **timeout** is reached.

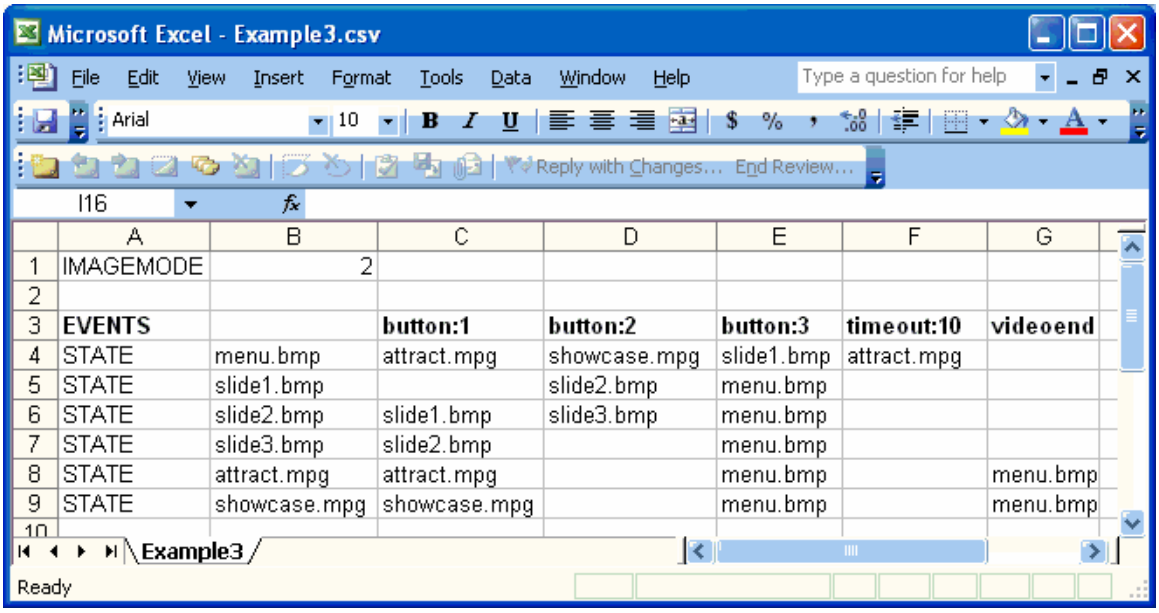
Example 2: *slideshow that advances only when a click / touch occurs*



Example 2 has only 1 event, the same **elo** event as Example 1.

- There's an elo event entry for each image that plays next image when there's a click or touch anywhere onscreen.
- There are no timeout events in this demo. Because example 2 only has an elo event, and no timeout events, you must click or touch the screen to advance the slideshow.

Example 3: *Video and Slide playback, using button input*



Microsoft Excel - Example3.csv

File Edit View Insert Format Tools Data Window Help — Type a question for help

Arial 10 — Reply with Changes... End Review...

I16

	A	B	C	D	E	F	G
1	IMAGEMODE	2					
2							
3	**EVENTS**		**button:1**	**button:2**	**button:3**	**timeout:10**	**videoend**
4	STATE	menu.bmp	attract.mpg	showcase.mpg	slide1.bmp	attract.mpg	
5	STATE	slide1.bmp		slide2.bmp	menu.bmp		
6	STATE	slide2.bmp	slide1.bmp	slide3.bmp	menu.bmp		
7	STATE	slide3.bmp	slide2.bmp		menu.bmp		
8	STATE	attract.mpg	attract.mpg		menu.bmp		menu.bmp
9	STATE	showcase.mpg	showcase.mpg		menu.bmp		menu.bmp
10							

Example3

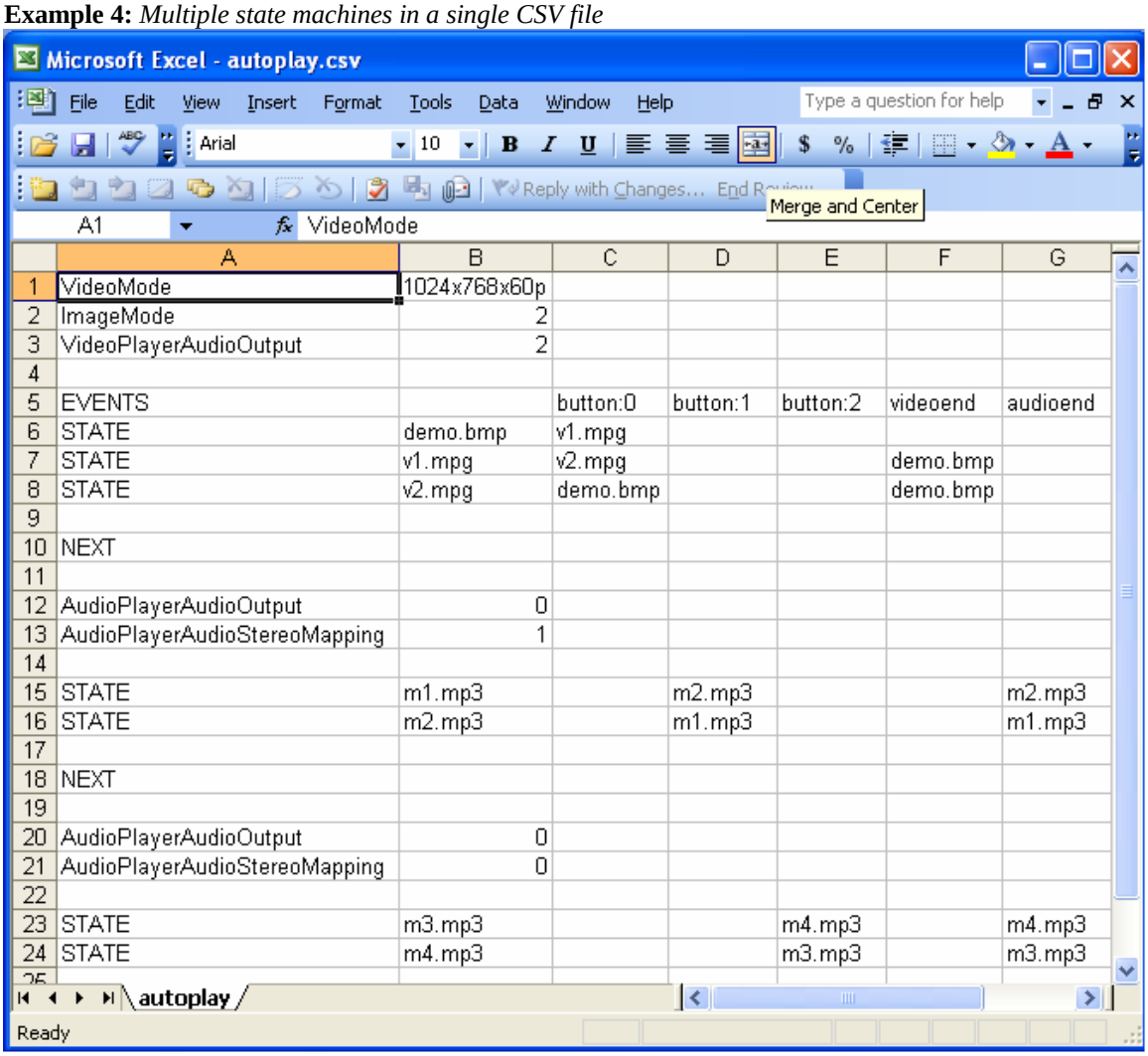
Ready

Example 3 contains five events, three **button** events and a **timeout** and **videoend** event. This demo plays two videos and 3 images.

When this demo starts, **menu.bmp** is displayed since it's in the first STATE position.

- If the menu is being displayed, pressing button 1 starts the attract video; pressing button 2 starts the showcase video; pressing button 3 starts the slideshow. If no button is pressed, the attract video starts automatically after 10 seconds.
- If the attract or showcase video is playing, pressing button 1 restarts it.
- If the attract or showcase video is playing, pressing button 3 returns to the menu. Since there's no entry under button 2 for either video, button 2 is ignored.
- When either the attract or showcase video ends, the menu is displayed, as indicated under the videoend event.
- For the slideshow, button 2 advances to the next slide, and button 1 goes back to the previous slide. Button 3 returns to the menu.

Example 4: *Multiple state machines in a single CSV file*



	A	B	C	D	E	F	G
1	VideoMode	1024x768x60p					
2	ImageMode	2					
3	VideoPlayerAudioOutput	2					
4							
5	EVENTS		button:0	button:1	button:2	videoend	audioend
6	STATE	demo.bmp	v1.mpg				
7	STATE	v1.mpg	v2.mpg			demo.bmp	
8	STATE	v2.mpg	demo.bmp			demo.bmp	
9							
10	NEXT						
11							
12	AudioPlayerAudioOutput	0					
13	AudioPlayerAudioStereoMapping	1					
14							
15	STATE	m1.mp3		m2.mp3			m2.mp3
16	STATE	m2.mp3		m1.mp3			m1.mp3
17							
18	NEXT						
19							
20	AudioPlayerAudioOutput	0					
21	AudioPlayerAudioStereoMapping	0					
22							
23	STATE	m3.mp3			m4.mp3		m4.mp3
24	STATE	m4.mp3			m3.mp3		m3.mp3

Example 4 contains three state machines.

The first state machine specifies the video mode, image mode, and audio output for video. In this case, the keyword VideoPlayerAudioOutput directs the BrightSign to output stereo PCM via the SPDIF audio connector. This machine uses button 0 and the video end event to control the videos and the image.

Note the **NEXT** keyword on line ten; it signals the beginning of another state machine.

The second state machine uses analog audio output through the Audio 2 connector. It uses button 1 and the audio end event to control the audio playback.

The **NEXT** keyword on line 18 indicates the start of the third state machine.

The third state machine uses analog audio output through the Audio 3 connector. This state machine uses button 2 and audio end events to control its audio playback.

Rollover Regions using State Files (CSV)

In the previous section, "Using Excel to Create Interactive Signs", you learned that you can create touchable (clickable) areas on the screen using **elo** (**elor)** and **eloc** keywords in the EVENTS row of a CSV file. You can then use these regions like buttons to playback content. Any touch region can be setup as a rollover region. Rollover regions allow you to create more dynamic and interactive displays that accept mouse and touch input. A rollover region has an ON and an OFF image. This allows your display to not only respond to clicks or touch input, but also to the movement of the cursor. Whenever the mouse cursor is within that region the ON image is displayed and the rest of the time the OFF image is displayed. This allows buttons to highlight as the mouse cursor moves over them.

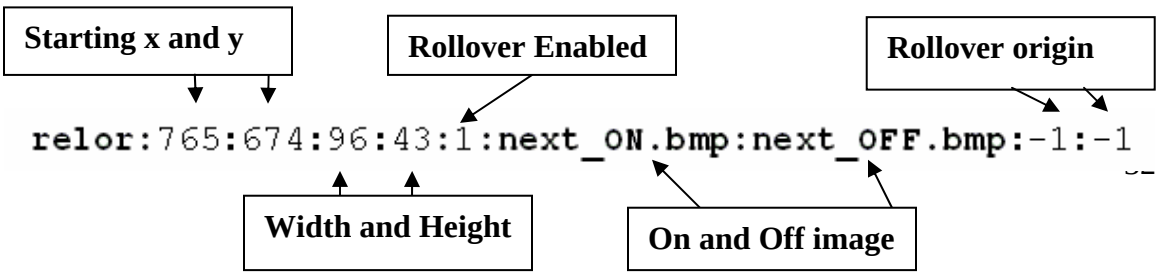
Rollover Events

Use either of the following keywords in the events row to setup a rollover region:

- **Relor** – you define a rectangular rollover region using this format:
  - relor:x:y:width:height:enable_rollover:on_bitmap:off_bitmap:x_origin:y_origin
  - **x:y** - represents the starting position, where 0,0 is in the upper left
  - **width, height:** width and height of the touch area in pixels
  - **enable_rollover:** 1 to enable rollovers, 0 to disable rollovers
  - **on_bitmap:** bitmap image name. Bitmap displayed when the mouse is within the defined region
  - **off_bitmap:** bitmap image name. Bitmap displayed when the mouse is not within the defined region.
  - **x_origin,y_origin:** rollover origin - the starting coordinates for the rollover image. Set to -1, -1 to keep the default origin (same as the touch area).

- **Reloc** – you define a circular rollover region using this format:
  - reloc:x:y:radius:enable_rollover:on_bitmap:off_bitmap:x_origin:y_origin
  - **x, y:** center of the touch area
  - **radius:** radius of the touch area in pixels
  - **enable_rollover:** 1 to enable rollovers, 0 to disable rollovers
  - **on_bitmap:** bitmap image name. Bitmap displayed when the mouse is within the defined region
  - **off_bitmap:** bitmap image name. Bitmap displayed when the mouse is not within the defined region.
  - **x_origin,y_origin:** rollover origin - the coordinates for the center of the rollover image. Set to -1, -1 to keep the default origin (same as the touch area).

Example 4: *rectangular rollover region*



This is an example of what you would enter in the CSV file to define a rollover region. In this example:

- Starting position is: 765, 674
- The touch region is 96 pixels wide and 43 pixels high
- The rollover region is enabled
- The on and off images are: next_on.bmp and next_off.bmp.
- The rollover images are displayed in the same location as the touch region

Normally, the on and off image have the same dimensions as touch region (width, height), and the same starting position (x,y). The rollover origin (x_origin, y_origin) allows you to specify a different starting position. You can use this feature if your on and off images are larger than the touch region you use to activate the on and off images. Or, you may have a menu with thumbnail images on the left side, and you want a larger image to be displayed on the right side of the screen when the cursor is over a thumbnail.

The following CSV example plays a simple slideshow with 3 images. It has two rectangular rollover region events and one timeout event. Examples 5.1 and 5.2 are the same as Example 5, but expanded so it's easier to read.

- If nothing is clicked or touched, the slide show will start automatically after the 10 second timeout, and then return to the menu.
- Similar to a standard elo entry, you'll notice that clicking in the first touch area, advances to the next slide until you reach menu.bmp.
- Clicking in the second touch area takes you back to the previous slide.
- In example 5.1, we see clearly that each of the 3 images has a section that's 96 pixels long by 43 pixels tall that's defined as a rollover region.
    - When the pointer isn't over this area, next_notlit.bmp is diplayed
    - When the point is over this area, next_lit.bmp is displayed
- In example 5.2, we see that each of the 3 images has a section that's 95 pixels long by 43 pixels tall that's defined as a rollover region.
    - When the pointer isn't over this area, back_notlit.bmp is displayed
    - When the pointer is over this area, back_lit.bmp is displayed

Example 5: *slideshow, with next and back buttons that light up when the mouse is over them*

	A	B	C	D	
1	IMAGEMODE	2			
2					
3	EVENTS		relor:765:674:96:43:1:next_lit.bmp:next_notlit.bmp:-1:-1	relor:643:674:95:43:1:back_lit.bmp:back_notlit.bmp:-1:-1	timeout:10
4	STATE	menu.bmp	slide1.bmp		slide1.bmp
5	STATE	slide1.bmp	slide2.bmp	menu.bmp	slide2.bmp
6	STATE	slide2.bmp	menu.bmp	slide1.bmp	menu.bmp
7					

Example 5.1:

	A	B	C	
1	IMAGEMODE	2		
2				
3	EVENTS		relor:765:674:96:43:1:next_lit.bmp:next_notlit.bmp:-1:-1	relor:6
4	STATE	menu.bmp	slide1.bmp	
5	STATE	slide1.bmp	slide2.bmp	menu.l
6	STATE	slide2.bmp	menu.bmp	slide1.
7				

Example 5.2:

	D	E
bmp:-1:-1	relor:643:674:95:43:1:back_lit.bmp:back_notlit.bmp:-1:-1	timeout:10
		slide1.bmp
	menu.bmp	slide2.bmp
	slide1.bmp	menu.bmp

Scheduling Playback (HD2000 only)

Videos, still images, playlists, and CSV files can be played back at scheduled times using the autoschedule.txt file. This file contains a series of date/time specifications, followed by a specification of what to play when the date/time event occurs.

The syntax of the date/time specification is as follows:
Year "/" Month "/" Day "-" Day of Week "-" Hour ":" Minutes

For example, a date/time specification for December 23, 2007 at 3:20 PM is defined as:
2007/12/23-*-15:20

In this example, the Day of Week specification parameter is a wildcard defined with an asterisk. All date/time parameters not used must be defined as wildcards using an asterisk (*).

Valid syntax for Date/Time parameters:

Year:	"2000".."2038" (4 digit integer from 2000 to 2038)
Month:	"01".."12" (2 digit integer from 01 to 12)
Day:	"01".."31" (2 digit integer from 01 to 31)
Day of Week:	"Sun" "Mon" "Tue" "Wed" "Thu" "Fri" "Sat"
Hour:	"00".."23" (2 digit integer from 00 to 23)
Minute:	"00".."59" (2 digit integer from 00 to 59)

In the autoschedule.txt file, the line following a date/time specifications must be a csv file, playlist file, video file or image file. When the date/time event occurs, whatever is currently playing is stopped and the new file begins execution.

The following is an example autoschedule.txt file:

//*-MON-01:00
weekday.mpg

//*-SAT-01:00
weekend.mpg

In this example, the weekday.mpg file is launched on Monday at 1:00 AM and plays continually during the week; on Saturday at 1:00 AM, the file weekend.mpg is launched and plays until Monday at 1:00 AM.

Another example autoschedule.txt file:

2007/12/15-*-15:20
image.png

2007/12/25-*-16:16

music.bsp

2008/1/*-WED-16:00
video.mpg

2008/1/*-*-07:00
breakfast.bsp

2008/1/*-*-14:00
demo.csv

In this example, the image “image.png” will be displayed starting on December 15, 2007 at 3:20 PM. On December 25, at 4:16 PM, the playlist “music.bsp” will start. The next specification indicates that “video.mpg” will start playing at 4:00 PM every Wednesday in January, 2008. The playlist “breakfast.bsp” will start playing at 7:00 AM during January, 2008. Finally, “demo.csv” will be launched everyday in January, 2008, at 2:00 PM.

Note that blank lines can be added (as above) to autoschedule.txt files to improve readability. Additionally, “REM” lines can be added to annotate these files.

Scripts

BrightSign can be used without any knowledge of scripts to accomplish common goals such as playing looping video or playing a slideshow (see the prior section in this manual “Playing Media”). It’s easy to control your digital signs or kiosks with a USB keyboard or remote using a simple script. In addition, using CSV files you can create more sophisticated signs. But by adding scripts, you can create the most interactive digital signs or kiosks. For a fee, Roku can help you write scripts. Contact brightsignsales@rokulabs.com for more information.

A script is a simple program that controls behaviors, like video playback, based on events such as button presses. Scripts are also commonly used on PCs and Web Sites. For example, Microsoft Office “macros” are often Basic Scripts. Adobe Flash apps often rely on ActionScripts for their logic, and web pages are often driven by scripts written in PHP, Perl, or JavaScript. If you don’t have scripting skills in-house, IT and Web developers can be contracted to write scripts for you. Roku can also be contracted to develop custom scripts and applications (contact brightsignsales@rokulabs.com).

BrightSign has a built in scripting language (called BrightScript), similar in many ways to the Basic found in Microsoft Office, or Windows XP “Windows Script Hosting” VBScript.

Scripts can be as simple as one or two lines, and as complicated as thousands of lines.

Hello world

Here is how to create your first, very simple script

1. Using a text editor create a file called “hello.bas” with this line:

```
print “hello world”
```

2. Save “hello.bas” into the root directory of a CF card
3. Insert the CF card into BrightSign
4. Connect the serial port of BrightSign to you PC using a null modem cable, and run hyperterminal (see the “Shell” section of this manual for more details).
5. Power up BrightSign.
6. Wait for it to boot, and for the “Roku>” shell prompt to appear
7. type: `script /hello.bas`
8. You should see “hello world” on your terminal window.
9. If you want the script to autorun upon boot, rename it “autorun.bas”

Roku Objects

As well as the BrightScript language itself, BrightSign relies upon a library of “Roku Objects” to expose the hardware to the scripting language. Each Roku Object provides access to certain capabilities. For example, there is a roVideoPlayer object and a roTouchScreen object. To create an object in a script, the CreateObject() function is used.

Refer to the following two manuals when writing scripts:

- Roku BrightScript Reference -- this describes the actual scripting language
- Roku Object Reference -- this describes the Roku Objects, such as “roVideoPlayer”

Examples

A good way to learn BrightScript is to look at or modify existing scripts. You can find links to example scripts at www.rokulabs.com/brightsign. The BrightSign Demo “autorun.bas” is an example of a more sophisticated script.

You can also test BrightScript commands using the shell. Here is an example:

```
Welcome to the Roku Shell version 1.1.28 Release
Type '?' for help or 'help <command>' for help on <command>.

Roku> script
BrightScript Micro Debugger.
Enter any BrightScript statement, debug commands, or HELP.

BrightScript> print "hello"
hello
BrightScript> ? "hello"
hello
BrightScript> ?5+6
```

```
 11
BrightScript> ?type(5)
rotINT32
BrightScript> if 1=1 then print"1 does equal 1!" else print "strange!"
1 does equal 1!
BrightScript> ?right("abcdef",3)
def
BrightScript> exit
Roku>
```

Registry (HD2000 only)

The BrightSign HD2000 has an I2C EEPROM with 2K bytes of memory that supports the storage of persistent settings. The BrightSign software registry object is how BrightScript reads and writes data to this eeprom. The purpose of the registry is to enable customers to persist usage statistics and other small amounts of information that may change regularly

Individual items are stored in the registry through the use of registry keys and values. For example, there might be a registry key named “NumberOfButtonPresses”; the corresponding registry value would be used to count the number of times a button is pressed. Registry data can be organized into different registry sections if desired.

CSV files support writing to the registry, deleting from the registry, and displaying the contents of the registry – see the Using Excel to Create Interactive Signs section earlier in the document. The contents of the registry can also be displayed from a playlist – see the Playlist Commands section earlier in the document. In addition, the registry can be accessed by a custom script using the roRegistry object (see the BrightSign Object Reference Manual for additional information).

The number of registry keys and values that can be used is dependent on a number of factors, including the number of sections, the number of keys/values per section, and the size of the keys and values. In typical usage, there’s room in the registry for about 50 keys.

GPIO Control Port

The BrightSign has a DB25 General Purpose Input Output (GPIO) Port. This port has 12 inputs that can be directly connected to buttons or switches, and 6 outputs that can directly drive LEDs. To use this port you must create a simple but custom cable that connects a DB25 connector to your buttons or LEDs. See the BrightSign Hardware Manual for more information and example circuits. Roku has a test button/LED board that you can purchase to evaluate the GPIO port and test your scripts that read the buttons or set the LEDs.

Here is an example script that will flash an LED for half a second each time a button is pressed on the BrightSign button/led board.

```
print "BrightSign Button-LED Test Running"
```

```
p =   CreateObject("roMessagePort")
tmr = CreateObject("roMessagePort")
gpio =  CreateObject("roGpioControlPort")
gpio.SetPort(p)

event_loop:
   msg=wait(0, p)
   if type(msg)<>"roGpioButton" then event_loop
   butn = msg.GetInt()
   if butn > 5 then event_loop
   gpio.SetOutputState(butn+17,1)
   print "Button Pressed: ";butn
   msg = wait (500, tmr)
   gpio.SetOutputState(butn+17,0)

   clear_events:
       msg=p.GetMessage():if type(msg)<>"rotINT32" then clear_events
   goto event_loop
```

Learning More and Support

The following resources can help you get the most from your BrightSign:

1. Visit www.rokulabs.com/brightsign
2. Refer to each of the different BrightSign documents available (listed at the start of this manual).
3. Visit the BrightSign community Forums to discuss projects with other customers and Roku engineers: www.rokulabs.com/forums
4. Contact brightsignsales@rokulabs.com for help with custom projects

Troubleshooting

If you are still having problems after trying troubleshooting tips, please contact customer support at www.rokulabs.com/brightsign for more assistance.

Autoplay & General Troubleshooting

Autoplay.mpg does not play automatically

1. Verify autoplay.mpg is spelled correctly
2. Verify that the autorun.bas v1.1 or later and autoplay.mpg are both in the root of the CF card
3. Verify you do not have an "autoplay.csv" or "autoplay.bsp" in the root of the CF card. There should only be one autoplay file in the root of the CF card.
4. Check if your PC is set to show file extensions. If your PC is not set to show file extensions, and you rename the file "autoplay.mpg", you may have accidentally named the file "autoplay.mpg.mpg."
5. Use the "BrightSign Shell" to test video playback.
6. See "General Playback Problems" for more troubleshooting help.

Autoplay.bmp does not play automatically

Naming your bitmaps autoplay1.bmp, autoplay2.bmp, etc is not supported under autorun.bas version 1.1 or later. Use a playlist to play images in a loop. See "Playlist" section for details.

General Playback Problems with the BrightSign

If you're having general problems playing back content with the BrightSign, try the following:

1. Download and verify "BrightSign Demo" plays. If you are also having playlist problems, test the Playlist Demo.
2. Verify content is correct format and matches specifications
   a) Audio – mp3, 48Khz sample rate (or 44.1KHZ sample rate on the HD2000), or wav with 16 bit linear PCM data on the HD2000.
   b) Video – standard MPEG-2, with 48Khz audio layer (audio layer not required on the HD2000)
   c) Image – Bitmap, 8-bit, 24-bit, or 32-bit only or PNG/JPEG files on the HD2000
3. Reformat your CF card as FAT32.
4. Copy your content including any scripts, .csv files, or playlists to the card again.
5. Download and install the latest BrightSign software. Reinstall the software if you are already using the latest. See "Device Software and Upgrading" for details.
6. Try a different brand and/or size card with the BrightSign.

7. Re-create content with different software, if possible.
8. Contact Roku Customer Support if you are still having problems.

Video output includes artifacts, quality lower than expected

The format of video content should be chosen to best match the properties of the intended display resolution. Most of the video output modes are progressive and so for the highest quality video reproduction, video files should be MPEG-2 encoded as progressive files. This is because to present an interlaced MPEG-2 file on a progressive display, the video must be digitally de-interlaced which can introduce video artifacts e.g. diagonal lines look less smooth than if the file was encoded as progressive.

Additionally, choose the highest quality output possible.

Video output is red

On an HD2000, if the video is configured for component / HDMI out, the video will appear red if a monitor is connected to a VGA monitor. The video output defaults to component/HDMI out.

When using a VGA monitor, ensure that the video mode is set to 1024x768.

For playlists (autoplay.bsp), add the following line:
videomode 1024x768x60p

For CSV files (autoplay.csv), add the following line:
VIDEOMODE 1024x768x60p

No Audio during playback

The Autoplay script defaults to the Audio 1 output. This is the default output for your Playlist, CSV, and Autoplay.mpg files.

1. Verify your audio cable is firmly connected to the Audio 1 output on the HD600
2. Check the volume on your powered speakers / AV receiver.
3. Verify the AV receiver is set to the correct input.
4. If you set the volume or audio output in your playlist, double-check the settings used.
5. If you are using a custom script (including a modified autorun.bas), verify the audio output mode you've set matches the outputs you are using on the back of the BrightSign. See "roAudioPlayer" and "roVideoPlayer" sections in the BrightSign Object Reference for more details.

Playlists

Download the latest Autoplay script from www.rokulabs.com/brightsign.

Playlist support requires autorun.bas v1.1 or later.

1. Verify autoplay.bsp is spelled correctly
2. Verify that the autorun.bas v1.1 or later and autoplay.bsp are both in the root of the CF card
3. Verify you do not have an “autoplay.csv” or “autoplay.mpg” in the root of the CF card. There should only be one autoplay file in the root of the CF card.
4. Check if your PC is set to show file extensions. If your PC is not set to show file extensions, and you rename the file “autoplay.bsp”, you may have accidentally named the file “autoplay.bsp.txt.”
5. Use the “BrightSign Shell” to test playlist playback. You’ll be able to see any errors.
6. See “General Playback Problems” for more troubleshooting help.

State Files (CSV)

Download the latest Autoplay script from www.rokulabs.com/brightsign.
Playlist support requires autorun.bas v1.1 or later.

1. Verify autoplay.csv is spelled correctly.
2. Verify that the autorun.bas v1.1 or later and autoplay.csv are both in the root of the CF card
3. Verify you do not have an “autoplay.bsp” or “autoplay.mpg” in the root of the CF card. There should only be one autoplay file in the root of the CF card.
4. Verify your content is in the root of the CF card, not in any subfolders.
5. Verify the spelling of file names in the autplay.csv file. If you pasted the file names into the csv file, check for any leading or trailing spaces in the name.
6. Verify the case and spelling of any keywords in autoplay.csv
7. Use the “BrightSign Shell” to test playlist playback. You can see any errors reported by the autoplay script.
8. See “General Playback Problems” for more troubleshooting help.

BrightSign and Accessories

The BrightSign isn’t booting up properly / Random lights lit

When the BrightSign starts up, it cycles through the Connector Activity and Video Mode lights. The only two lights that should remain lit at the end of the startup process are the Video Mode light, normally 1024x768, and the Power light.

1. Download and reinstall the BrightSign software. See “Device Software and Upgrading” for details.
2. If reinstalling the BrightSign software doesn’t fix the problem, please contact Roku Customer Support at www.rokulabs.com/brightsign.

BrightSign Software Update Fails

1. Verify that the update file is named update.rok
2. Verify the update file is in the root of the CF card, not in a sub-directory
3. Try reformatting the CF card as FAT32
4. Try using a different CF card
5. Try updating from the BrightSign Shell. Please see "Device Software and Upgrading" for details.
6. Contact Roku Customer Support, www.rokulabs.com/brightsign., if you are still having problems.

Compact Flash card works for playback, but can't update BrightSign

Your CF card might not be compatible with the "safe mode" software on the BrightSign. If that's the case, you can try manually updating the software from the BrightSign Shell or using a different CF card to do updates. Please see "Device Software and Upgrading" for more details.

Compact Flash card is not recognized by the BrightSign

If the BrightSign doesn't see any content on your CF card, that card might not be compatible with the BrightSign.

1. Reformat your CF card as FAT32.
2. Try a different brand and/or size card with the BrightSign.
3. Once you find a working card, download and install the latest BrightSign software. See "Device Software and Upgrading" for details.
4. Try the problem card again once the BrightSign's software has been updated
5. Contact Roku Customer Support, www.rokulabs.com/brightsign., if you are still having problems.

Video Selector switch does not work

The video selector switch is not currently implemented on the HD2000.

On an HD600, the selector switch will not work if the video mode is set manually using a script, playlist, or .csv file. To verify if this is the case, startup the BrightSign without a CF card inserted. If the video selector switch doesn't work when no CF card is inserted:

1. Download and reinstall the BrightSign software. See "Device Software and Upgrading" for details.
2. If the selector switch still isn't working, contact Roku Customer Support, www.rokulabs.com/brightsign.

USB Mouse is not responding

The BrightSign works with many different brands of USB mice. Before troubleshooting the mouse, please download the BrightSign Demo, and test if your mouse works with the demo. If the mouse works with the demo, then the problem is with your script or the coordinates used in the .csv file.

1. Unplug the mouse from the BrightSign, and reconnect
2. If you are using a wireless mouse, reconnect the wireless transmitter, and replace the batteries in the mouse
3. Try connecting your mouse to the BrightSign using a USB hub
4. Your mouse might not be compatible with the BrightSign. Try a mouse from this list of tested mice.

USB Keyboard is not responding

The BrightSign works with many different brands of USB keyboards. To use a keyboard, you must write a custom script. The "buttons" supported by the autoplay.csv are for input from the GPIO port.

1. Disconnect the keyboard from the BrightSign and reconnect
2. Try connecting your keyboard to the BrightSign using a USB hub
3. Try a different USB Keyboard.
4. Test

Elo Touch Screen coordinates aren't working properly

If the Elo coordinates appear to be reversed or flipped when using autoplay.csv, try the following:

1. Download version 1.1 or later of the Autoplay Script
2. Open autoplay.csv and add this keyword, "FLIPELO", to the top of the csv file

```
FLIPELO
IMAGEMODE              2

EVENTS                       elo:110:180:355:210
```

3. Copy the updated autoplay.csv and autorun.bas to your CF card
4. Insert the CF card into your BrightSign and add power.

Appendix

Selecting Compact Flash Cards

Summary
When using a BrightSign Product, your content and scripts are loaded onto a Compact Flash (CF) card. Although the CF interface is standardized, there are a wide range of performance, compatibility, and reliability issues with different CF card brands and models. Roku strongly recommends that you use "industrial rated" CF cards with BrightSign. We have tested "industrial rated" CF cards from the following brands:

- Transcend
- PQI
- Simpletech

Avoiding CF Card Corruption due to "Read Fatigue"
Failure to use an Industrial CF card can result in files on the CF card becoming corrupt after a period of time. CF cards use NAND flash memory chips. These chips are rated at a certain number of "block reads" before a particular block may have a read error. Different types of NAND flash are more susceptible to these read errors than others. For example, "multi-level" flash chips are much more susceptible to this read issue than "single level". CF card controllers (the chip inside the CF card that controls the flash memory and interfaces with the CF connector) are responsible for eliminating these read errors as well as write errors through various strategies. These include "wear leveling" when writing, automatic error correction on read, re-writing a block that is becoming susceptible to read errors, and moving blocks around in the NAND chip that are repeatedly read over and over. If the CF card is designed correctly for repeated use, the CF card user will never notice read or write issues. However, not all CF cards are designed this way: in particular, certain models of consumer CF cards are not designed for repeated read use.

In a typical BrightSign application, there might be an "attract video loop". If this video is fairly short, then a small number of flash blocks will be repeatedly read. Over a month or so, the number of times the attract loop is read can be enough to trigger the "fatigue read errors" if the CF card's controller chip is not designed correctly to handle and eliminate them. "Industrial rated" CF cards are designed with this high-repeated use in mind. Consumer CF cards may be designed in lower cost or quality ways that assume a digital camera is the typical usage scenario.

Speed
We have not found a modern CF card that did not have sufficient read speed for digital sign and kiosk applications with standard definition video. In general you want a card that can sustain over 3 megabytes per second read rates for standard video, with 4 megabytes per second easy to find and recommended. For Hi-Def video, we recommend a card that can sustain over 6 megabytes per second read rates. You can test a card's

speed with BrightSign by placing a file on it, and using the shell “readperf” command (see the BrightSign User Guide on shell instructions).

BrightSign Compatibility

BrightSign is compatible with most CF cards, but may not be 100% compatible with every model. Although there are standards for CF card interfaces, most chips are slightly different. We have tested with a large number of chips, and at this point have high compatibility. Earlier BrightSigns were less compatible, and if you have one of these you may want to upgrade your software. See the BrightSign user guide for a discussion of the two ROMs in a BrightSign (boot vs. run mode), and the upgrade and CF card compatibility issues with each ROM.

When deploying a group of BrightSigns into the field, Roku strongly recommends that you test the CF card with the BrightSign software version you will use before deployment, and insist your CF card duplicator not change CF brands or model numbers from the model number you have approved.

Generating HD Content for the BrightSign HD2000 using Final Cut Pro

From within Final Cut Pro:

- Open an existing sequence or create a new one
    - Add HD content to the sequence as needed
    - Ensure that audio tracks are enabled. If your project has no audio, add a Slug to one of the audio tracks
- Select the portion of the sequence to export
- Select File > Export > Using Compressor

In Compressor:

- Go to the Settings tab in the Settings window.
- Find Apple:Formats:MPEG-2:Program Stream and drag it onto your file in Compressor (where it says Drag Settings and Destinations Here).
- Select the Program Stream just dragged onto your file
- Select your file and check in the Inspector window that the resolution, frame rate and aspect ration are correct (click on the Encoder button in the Inspector window - second from the left). Most HD sources (HDCAM, HDV etc) will have set the resolution to 1440x1080. It'll need to be changed to "HD 1920x1080" if you're creating 1080i content or "720p" if you're creating 720p content by clicking on the change button to the right of the value of Video Format and then selecting the correct value from the pull down list. Frame rate and aspect ratio (usually 16:9) can be changed the same way.
- Once you're happy that the settings are correct, hit Submit

From the Finder:

- Select the movie to compress (typically the exported master program)
    - Ensure that the movie contains an audio track. To verify, open the movie in the QuickTime Player. Select Window->Show Movie Properties and check that a SoundTrack is displayed in the Properties window.
    - If there is no audio track, go to Final Cut Pro and add audio as described above.
    - To ensure that there is an audio track when exporting a movie from Final Cut Pro, be sure that "Include" is set to "Audio and Video" in the File > Export > QuickTime Movie > Save dialog.
- Launch Compressor
- Drag the movie file and drop it into Compressor (where it says Drag Settings and Destinations)
- Find Apple:Formats:MPEG-2:Program Stream and drag it onto your file in Compressor (where it says Drag Settings and Destinations Here).
- Select the Program Stream just dragged onto your file

- Select your file and check in the Inspector window that the resolution, frame rate and aspect ration are correct (click on the Encoder button in the Inspector window - second from the left). Most HD sources (HDCAM, HDV etc) will have set the resolution to 1440x1080. It'll need to be changed to "HD 1920x1080" if you're creating 1080i content or "720p" if you're creating 720p content by clicking on the change button to the right of the value of Video Format and then selecting the correct value from the pull down list. Frame rate and aspect ratio (usually 16:9) can be changed the same way.
- Once you're happy that the settings are correct, hit Submit

Device Software (Firmware) and Upgrading

The BrightSign is pre-programmed with two primary software images: a "Safe Mode" and a "Run Mode". The Safe Mode software image is write protected, and not changed once programmed by the factory. Its main purpose is to validate that the run mode rom image is not corrupt, and boot the unit. In addition, it can upgrade or re-program the main run-mode rom image.

The main run mode software image is stored on a flash rom and may be upgraded in the field. Roku is always working on improving the BrightSign software, so you should check our web site for possible upgrades. To upgrade the BrightSign run-mode rom, follow these steps:

1. Download the latest software from www.rokulabs.com/BrightSign
2. Unzip downloaded file and Copy the update to you CompactFlash card using your PC and a standard CompactFlash Card reader/writer.
3. Name the upgrade file either "update.rok" or "update_save.rok". "update.rok" will auto-delete after the upgrade is complete. This is useful if you want to distribute the update to the field on the same CF card that contains the display content. In this mode, after the first boot, the unit will no longer upgrade, but will continue to autorun any auto-run content on the CF card. If named "update_save.rok" the file will not be auto-deleted. This is useful for using the same CF card to upgrade multiple BrightSigns. On the HD2000, there are two additional options. If the file is renamed "update_newer.rok", the update contained in the file will only be applied if the version it contains is newer than the version that is currently running. The file is not deleted. If the file is renamed "update_different.rok", the update contained in the file will only be applied if the version it contains does not match the version that is currently running. The file is not deleted.
4. Insert the CF card into the BrightSign and Apply Power.
5. HD600
    a. Wait. If a display is connected it will be blank. After several seconds one of the LEDs on the front panel will start to flash. Once it completes flashing, the upgrade is complete. If you are using the "update.rok" method, the unit will then reboot as normal.

b. If you wish to verify the new software version, startup the BrightSign without a flash card inserted. If this update is older than version 1.1, use the BrightSign Shell and the "version" command. See the "BrightSign Shell" section for more details.

6. HD2000
    a. Wait. If a display is connected it will be blank. The LEDs on the front panel will be lit as the unit boots up. After some time, the red upgrade LED will light up.
    b. During the upgrade, the connector activity LEDs are turned on and off to indicate that the upgrade is in progress.
    c. If the upgrade file was named "update.rok", the file will be deleted and the unit will reboot.
        i. Note – if the card was formatted with NTFS or HFS+, then the update.rok file will not be deleted, the Video Mode LEDs will flash, and the unit will not reboot.
    d. If the upgrade file was named "update_save.rok", the Video Mode LEDs will flash. The file will not be deleted and the unit does not reboot.

The upgrade process is very safe. If for some reason it fails, for example a power failure in the middle of the update, the unit's "safe-mode" will still boot and will still perform upgrades as above.

Updating from the BrightSign Shell

BrightSign works with most CF cards, but is not compatible with all. See the "BrightSign Shell" section for more details on testing your CF cards and accessing the shell.

Because the BrightSign "safe mode" software is normally older than the "run mode" software, more CF cards are compatible with "run mode" than are compatible with the "safe mode". The effect of this is that "update.rok" file may not be detected on some CF cards. It is impossible to tell by a CF card's name, brand, or description if it is compatible. You must try it.

To manually update using the BrightSign Shell:

1. Insert your CF card with the update.rok file
2. Add Power to the Brightsign
3. Type the following at the Roku Prompt in the BrightSign Shell:

```
update update.rok
```

We are continuing to add support for more CF cards.

BrightSign Shell

BrightSign has a command line shell that can be accessed via the serial port. The shell allows you to type in commands to trigger tasks such as running scripts, playing video, trying different video modes, etc. It is not required that you use the shell. You can simply put your content on a CF card and configure it to auto play on power-up (see Playing Media with BrightSign).

You may find it useful to use the shell to test various options or for debugging. Follow these steps to access the shell:

1. Connect the DB9 connector on the BrightSign to your PC's Serial Port using a "null modem cable" (available at Frys, RadioShack, etc).
2. Run a "terminal emulator", such as HyperTerminal on Windows XP (Start | All Programs | Accessories | Communications | HyperTerminal.)
3. Configure the terminal for: 9600 baud on the HD600 or 38400 baud on the HD2000, 8 data bits, no parity, 1 stop bit, NO flow control, "TTY mode".
4. Insert a CF card into BrightSign with some test files on it (BMP graphics, 48KHZ mp3 files, MPEG-2 video, etc)
5. Connect speakers and/or a monitor to the BrightSign
6. Power-up the BrightSign
7. After a few moments you should see the BrightSign shell appear. It should also show you the software version number. Verify it is 1.0.72 or greater.

List of Shell Commands

With the shell running, type: help <enter> for a list of commands. Type the command on a line by itself for more information on a particular command.

Example Shell Commands: Playing Files & Changing Settings

All files on the CompactFlash card should be referenced with a "/" in front of the filename.

For a Directory of the cards contents use:

```
dir /
```

To show the contents of a sub-directory named images, use:

```
dir /images
```

To play a video named "myvideo.vob" on the CF card, enter this command:

```
videoplay /myvideo.vob
```

To play a video with a space in its name, my video.vob:

```
videoplay "/my video.vob"
```

To change volume (0 - 100) to 75% when playing back a video or audio use:

videoplay volume 75
audioplay volume 75

To change the videomode use the videomode command. Example:

```
videomode 1024x768x60p
```

To display an image, use the imageplay command. Example:

imageplay slide1.bmp

To specify an image mode, use the imageplay command with mode setting. Examples:

imageplay slide1.bmp 0
imageplay slide1.bmp 1

The current version of the BrightSign software does not support "hot swap" of CompactFlash cards. You must put in the CF card before applying power.

Using SD Cards (HD2000 only)

To list the contents of a CF card, enter any of the following:
dir
dir cf:/
dir ata:/

To list the contents of an SD card, enter:
dir sd:/

To explicitly refer to a card from within a script, use "CF:/" for compact flash cards or "SD:/" for SD cards.

When an autorun.bas script is run, its current drive is set to the drive containing autorun.bas, so references to files on the same card do not need the CF:/ or SD:/ prefixes.

Testing Compact Flash Cards

BrightSign works with most CF cards, but is not compatible with all CF cards. You can test your card by using the "dir /" shell cmd, and/or using the shell cmd "readperf". Examples:

```
dir /
```

```
readperf bigfile.mpg
```

If the “dir” command fails, you’ll get the following result:

```
Roku> dir /
opendir[/]
Error: can't open dir
```

If the “readperf” command fails, you’ll get the following result:

```
Roku> readperf bigfile.mpg
SET FEATURES transfer mode SUCCEEDED

Unable to open file ATA:\autorun.bas
```

Testing Scripts, Playlists, and .CSV

An efficient way to test playback is to use the BrightSign Shell. By using the shell, you can see “debug output”. From the BrightSign Shell you can start your script manually and view output from the script. You can start a script from the shell by using the “script” command. Follow the directions above to connect the BrightSign to your computer, and access it using Hyperterminal.

As your script runs, you’ll see output in your terminal program. You’ll see a variety of output including mouse clicks or button presses, file names, audio and video settings, as well as any errors associated with incorrect use of functions or commands including the line numbers.

Testing Playlists

When your playlist contains the keyword “debug” the autorun.bas script provides detailed output to the BrightSign Shell during playback. To begin testing playlists, do the following:

1. Copy your content including playlist and autorun.bas to your CF card
2. Open your autoplay.bsp (main playlist file) and add the keyword: `debug`
3. Insert your CF card into the BrightSign, and add power

Testing Autorun and .CSV files

Running the autorun.bas in debug mode will provide detailed output to the BrightSign Shell while the script is running. To begin testing the autorun.bas script in debug mode, do the following:

1. Copy your content, including .csv file and autorun.bas to your CF card
2. Open autorun.bas and find the “debug=false” line near the top of the script

```
debug=false   'set to true to turn on debug prints; false to turn them off
```

3. Change to: `debug = true`
4. Rename autorun.bas file to "run.bas" for example, so it doesn't start automatically
   a) If you are testing a playlist, the main playlist must be named "autoplay.bsp"
   b) If you are testing a .csv file, it must be named "autoplay.csv"

5. Insert your CF card into the BrightSign, and add power
6. At the Roku prompt, type the following:

```
script run.bas
```

Testing Custom Scripts
To make troubleshooting your custom scripts much easier, we recommend adding checks and print statements like those found in the autorun.bas script. Here are a few examples:

- Add print statements telling you which portion of the script you are currently in
- Add print statements showing the type of event that just occurred
- Add print statements showing names of files read from the flash card
- Add print statements showing the coordinates of a mouse click

To begin testing your custom script:
1. Copy your content and script file to the compact flash card.
2. Name your script `run.bas`. You can call it anything, as long as it's not autorun.bas, so it doesn't start automatically.
3. Add a debug statement to the top of your script, like the autorun.bas. Now, we'll see output from the print statements you added to your script.

```
debug = true
```

4. Insert your CF card and Add power to the Brightsign.
5. At the Roku prompt, type the following:

```
script run.bas
```

Mult-run Script Limitations
Currently when a script terminates, it does not free resources allocated by Roku Objects. This means that if you use the "script" shell command to run a script, and the script exits, you may not be able to run it again successfully. You will need to reboot your BrightSign between script tests.

Linux Shell (HD2000 only)

The HD2000 has a Linux shell that can be accessed via the serial port.

To access the Linux shell, enter the following at the Roku prompt:
exit

When done using the Linux shell, enter the following at the Linux prompt:
reboot

Setting the Date/Time on BrightSign (HD2000 only)

BrightSign has a real time clock that can be used for scheduling content playback. In order to use BrightSign's scheduling features, the date and time on the BrightSign must be set.

Roku has written a script that provides an interactive means for setting BrightSign's date/time. The script is named SetDateTime.bas and is available at www.rokulabs.com/brightsign in the software download section.

Instructions for using this script:

1. Copy SetDateTime.bas onto the root directory of a CompactFlash card and rename it "autorun.bas".
2. Connect a USB keyboard or a GPIO button board to the BrightSign.
3. Insert the CF card into the BrightSign, apply power, and after a few moments, a screen will appear with a user interface to set the date/time.

Once BrightSign's date / time is set, the real time clock will keep it accurate even across power cycles.

Writing Boot Script & Logo to Rom

With BrightSign HD600 Software version 1.1, you can now change the startup logo or add a boot script on the HD600. Both are stored in the internal FLASH. Both the logo and boot script have a size limit of 200KB.

Using Your Own Boot Logo

1. Save your logo in the following way: BMP, 8 bit indexed, RLE encoded
2. Copy your logo to a CF card
3. Connect your HD600 to a PC using a null modem cable
4. Run a terminal emulation program such as Hyperterm
5. Insert the CF card and apply power
6. When a prompt is presented in the terminal program, type the following:

```
writebootfile logo.bmp boot.bmp
```

7. Assumes logo.bmp is the file name of your logo file on the CF card.
8. Reboot the system - your logo should appear

Using Your Own Boot Script

The script must be self contained. For example, it can't be used to play a playlist or csv file. You can use the boot script to play any content on your CF card (images, audio, and videos).

1. Save your boot script as myscript.bas
2. Copy your myscript.bas to a CF card
3. Connect your HD600 to a PC using a null modem cable
4. Run a terminal emulation program such as Hyperterm
5. Insert the CF card and apply power.
6. When a prompt is presented in the terminal program, type the following:

```
writebootfile myscript.bas autorun.bas
```

7. Assumes myscript.bas is the file name of your boot script.
8. Reboot the system

To restore the BrightSign to the original boot files, reinstall the BrightSign software using update.rok. See Device Software and Upgrading for step by step directions.

Note – this functionality is currently available on the HD600 only.

Widescreen vs. 4x3 Discussion

BrightSign can be used with widescreen or 4x3 displays. However the use of different display aspect ratios with the various BrightScreen options can often be confusing. In this section we try to clarify what you can expect.

Displaying Videos

Video and Images are currently treated a little differently. When a video is being played back, it is displayed by default in what is known in BrightSignese as "ViewMode 0". In other words, the video, whatever its aspect ratio, is scaled to fill the current output resolution (as set by the videomode command). What this means from a practical point of view is that you should author the content for 16x9 if you intend to use a 16x9 display modes (any component mode). If you intend to use one of the 4x3 display modes (any VGA mode, for example 1024x768), then you should author the content for 4x3. If you author your video for widescreen, but play it back at 1024x768 or 640x480, it will look "squished". It will not be automatically letterboxed like a DVD player would do. For example, the aquarium video that is included in the BrightSign demo (can be downloaded on our web site) is "widescreen". It will look squished when displayed in VGA, but will look correct when displayed in 720p.

Using a custom script or modified autorun.bas, you can force a 16x9 video to be letterboxed. Using "ViewMode 1" sets the video to be letterboxed and centered while maintaining aspect ratio. You can read more about "ViewMode" in the Object Reference Manual. Assuming a roVideoPlayer object called "video":

```
video.SetViewMode(1)
video.PlayFile("\Aq.mpg")
```

Note – SetViewMode is not currently implemented on the HD2000

Displaying Images

On the other hand, BrightSign displays images smarter. It knows the image resolution, and it assumes that all VGA resolutions are 4x3, and that all "component" resolutions (except 480p) are widescreen. BrightSign will then scale the image according to the imagemode setting. However, if you hook up a widescreen VGA monitor to the VGA port, you can expect to see distorted images. This is because 1024x768 is a 4x3 mode, and your widescreen monitor will stretch it. To avoid this confusion, for widescreen modes, it is best to use a component input monitor.

Supported Input Devices

With the 1.x version of the BrightSign the following devices can be connected to a BrightSign. If you have specific requests for device support, contact brightsignsales@rokulabs.com. We are constantly adding new device support and the ones we pick are customer driven.

- Elo Touch Screens
- Many brands of USB Keyboards
- Many brands of USB Speakers – HD600 only
- Many brands of USB Mouse or Trackballs
- Many brands of USB Bar Code Scanners (these appear as a keyboard).
- Quadravox Serial Button Board
- Any "GPIO" button, signal or LED (your custom board or led display). Roku has a test button board available for purchase on our estore.
- VGA Monitors
- Component-input TVs.
- NTSC and PAL monitors – HD600 only
- HDMI Monitors / TVs – HD2000 only
- High Definition Monitors/TVs
- Speakers/Amplifiers with stereo line-level inputs
- Audio playback devices with SPDIF optical digital audio input
- PC via Serial Port
- IR Remote Control using the "NEC" protocol. A compatible remote can be purchased from Roku.

USB Mice

The following mice have been tested with the Brightsign.

Type	Connection	Brand	Model	Name	Comments
Ball	Wired	Belkin	F8E841-BLK	MiniScroller Mouse	
Optical	Wireless	Belkin	F8E874	MiniWireless Optical Mouse	
Optical	Wired	Dynex	DX-OM20		
Optical	Wireless	Kensington	72216	PilotMouse Optical Wireless	requires usb hub
Optical	Wired	Logitech	M-BE58	First/Pilot Wheel Mouse	
Optical	Wired	Logitech	M-UV55A	Mini Optical Mouse	
Optical	Wired	Logitech	M-BT96a	Optical Mouse (3 buttons)	
Trackball	Wired	Logitech	T-BC21	Marble Mouse	
Optical	Wired	Logitech	M-BP86	MX310 Optical Mouse	requires usb hub
Optical	Wired	Logitech	m-bw112a	LX3 Optical Mouse	requires usb hub
Optical	Wireless	Logitech	RBS111	Cordless Mini optical mouse	requires usb hub
Optical	Wireless	Logitech	V200	Cordless notebook mouse	requires usb hub
Optical	Wireless	Logitech	LX5	Cordless Optical mouse	
Optical	Wired	Micorsoft	69H-00001	Comfort Optical Mouse 1000	